

CBD 기반의 XML 메시지 교환 시스템

김 호 진 (동국대학교 대학원 전자상거래학과, kimhojin@dongguk.edu)

강 성 배 (동국대학교 대학원 전자상거래학과, ksb2726@dongguk.ac.kr)

문 태 수 (동국대학교 상경대 정보산업학과, tsmoon@dongguk.ac.kr)

I. 서 론

오늘날 웹(Web)의 발전은 하루가 다르게 폭발적인 인기를 누리며 사용자에게 보급되기 시작했고, IT 산업계 전반에 걸쳐 많은 영향을 미쳐왔다. 특히 인터넷을 통한 새로운 비즈니스인 전자상거래(Electronic Commerce)에 많은 기여를 했다. 최근 이러한 인터넷 전자상거래의 혁명은 초기의 기업과 고객간(B2C)거래 보다는 기업간(B2B) 거래에 가장 큰 영향을 주고 있다. 이러한 웹이 발전하는 데에는 누구나 쉽게 만들고 사용할 수 있는 HTML의 단순함이 큰 몫을 담당했으나 Global Network 화되어 데이터의 빈번한 교환과 공유화가 이루어지고 웹 이용의 급증에 따라 전자상거래가 시간적, 공간적 제약을 극복한 세계 경제활동으로 지식, 정보기반시대의 새로운 패러다임으로 국가 경제, 사회의 변혁을 주도하는 수단으로 부각되고 있는 현 시점에서 사용자의 다양한 요구가 발생하게 되었다. 이런 패러다임들을 뒷받침 해주는 차세대 인터넷 언어로 각광받고 있는 것이 XML(eXtensible Markup Language)이다.

XML은 W3C(World Wide Web Consortium)에서 1996년 11월 XML을 새로운 표준으로 제안하고 1998년 2월 XML1.0 권고 안을 제정하였다. XML은 기존의 데이터를 보다 쉽게 표현, 교환, 저장할 수 있어서 무역, 금융, 멀티미디어, 전자상거래 등 다양한 분야에 응용되고 있다. 초기 XML은 문서관리 및 정보검색의 관점에서 응용분야가 확대되었고 그 중 기업간 전자상거래는 다자간 거래를 위한 통합된 구조 및 언어가 필수적이기 때문에, 기업간 전자상거래 표준으로 XML을 이용하는 비율이 급속도로 증가하고 있다.

이러한 추세에 따라 기업간 전자상거래를 위한 다수의 XML 기반의 B2B Application 및 Framework들이 개발되어 활용되고 있으며, 그 대표적인 예가 XML/EDI이다. XML/EDI는 기존의 데이터 전달에 초점을 둔 부가가치통신망(VAN) 기반의 EDI(Electronic Data Interchange)의 범위를 확대하여 전자상거래에 필요한 Framework를 제공한다는 면에서 전통적인 EDI와 다르다고 할 수 있다[3].

기업간 전자상거래 비즈니스 모델의 변화와 더불어 급변하는 기업의 비즈니스 환경과 정보시스템은 갈수록 거대하고 복잡, 다양해져 가고 있는데 반해 소프트웨어 기능에 대한 사용자들의 요구는 계속적으로 증가하고 있고, 수시로 수정/보완되어야 하는 외적 환경으로 소프트웨어의 개발 및 유지보수 비용은 급증하여 “소프트웨어 위기(Software Crisis)”란 용어가 생겨나게 되었다. 이러한 문제점을 인식함으로써 융통성(flexibility) 및 재사용(reuse)을 기반으로 하는 객체지향 기술은 소프트웨어 위기를 해결하기 위한 해결책으로 인식하고 있으며, 미래의 비즈니스 형태에 적합하도록 접근 방법을 제시하여준다[2].

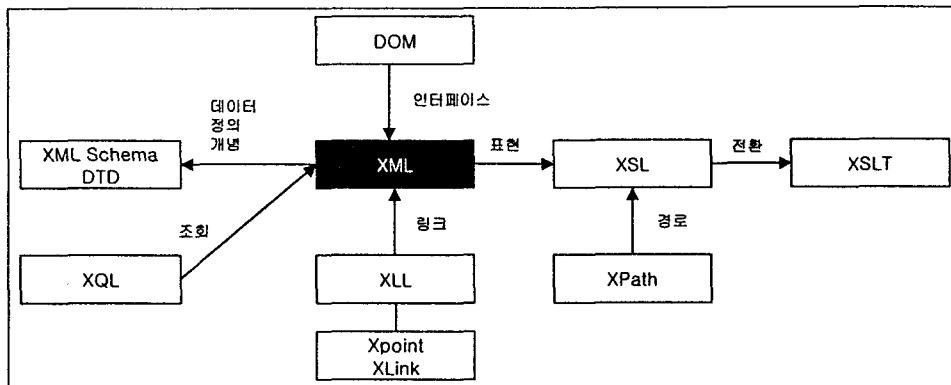
본 논문에서는 기업간 전자상거래 업무를 효과적으로 처리하기 위한 토대가 되는 XML 기반의 메시지 교환시스템을 CBD(Component-Based Development) 기반으로 구현한 결과를 제시한다. 그리고 컴포넌트기반의 메시지 교환시스템 구현을 통하여 컴포넌트의 재사용성을 높이고, 이식성과 확장성을 높임으로써 B2B 전자상거래를 활성화하고 소프트웨어 개발기법의 현장 적용능력을 향상하고자 한다.

II. 관련연구

2.1 XML

XML은 구조화된 문서의 구조를 정의하는 마크업 언어(Markup Language)라 할 수 있다. XML은 1996년 W3C에 의해서 처음 제안되었으며, 태그를 자유롭게 정의할 수 있는 SGML(Standard Generalized Markup Language)의 장점과 인터넷상에서 손쉽게 하이퍼미디어 문서를 제공할 수 있는 HTML(HyperText Markup Language)의 장점을 수용하여 만들어졌다.

XML은 <그림 1>에서 살펴볼 수 있듯이 다양한 구성요소를 통해 이루어져 있다. 문서 구조 및 내용은 XML 태그를 통해 나타내고, 실질적인 표현은 XSL(eXtensible StyleSheet Language)이 담당하고 있다. 업체간 상이한 문서 표준을 상호 호환시키기 위해 데이터를 정의하고 의미를 부여해주는 역할은 DTD(Document Type Definition)와 XML Schema가 맡고 있으며, 여러 업체와의 공통된 인터페이스를 유지하기 위한 요소로는 DOM(Document Object Model)이 사용되고 있다. XLL(XML Linking Language)은 인터넷 환경을 고려하여 다양한 관련 정보와 연관시킨 하이퍼미디어(Hypermedia)시스템을 구축하기 위해 만든 것이다[8].



<그림 1> XML 구성요소 및 역할

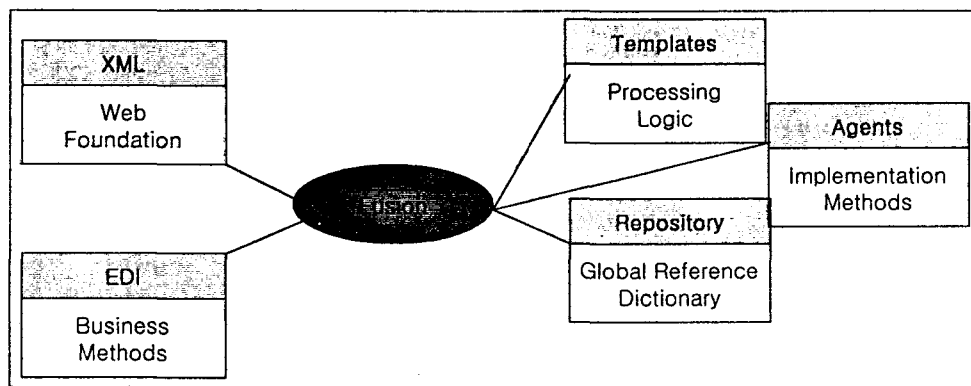
2.2 XML/EDI

일반적으로 기업이나 공공기관에서 EDI를 도입하게 되면 거래 처리시간 단축, 업무 부대비용 감소, 업무오류 방지, 물류비 절감, 이미지개선, 인력절감 등의 효율성 증대 효과를 기대하게 된다. 이러한 기대를 가지고 여러 기업이나 공공기관들이 EDI를 도입하였지만, 실질적으로 얻은 효과는 그리 크지 못하였고 EDI의 이론적인 도입효과에 비하여 EDI의 실질적인 활용도는 매우 낮은 것으로 나타나고 있다. 1990년대 후반부터 급속히 발전한 인터넷 관련 기술의 힘을 빌어 전통적 EDI의 문제점들을 보완하려는 차세대 EDI 기술에 대한 연구가 급속히 연구되어 왔다. 이러한 연구들 중에 EDI를 통한 효율성 제고 및 EDI 도입/운영비용의 절감 등의 관점에서 가장 주목받고 있는 차세대 EDI 기술은 Open EDI와 인터넷 EDI라고 할 수 있다. Open EDI는 개방형으로 구성되어 있어 기존 타 업무와 EDI시스템간의 연계를 통한 업무처리의 효율성을 높일 수 있으며, 인터넷 EDI는 메시지 전송을 위하여 비싼 VAN을 이용할 필요 없이 인터넷에 연결만 되어 있다면 무료로 이용할 수 있다는 점 때문에 자금력이 부족한 중소기업에까지 쉽게 확산 가능한 대안으로 주목받고 있다.

XML/EDI는 이러한 두 가지 EDI 시스템의 장점을 모두 가지고 있는 차세대 EDI 시스템

이라고 할 수 있다. XML/EDI는 HTML 적용 이후 인터넷 기술을 기업간 전자상거래 영역으로 확산하는 데에 중요한 기반을 제공하고 있으며, XML 기술의 메시지 교환시스템 적용으로 전통적인 EDI 시스템의 문제점을 해결하는 차세대 EDI라고 할 수 있다. XML/EDI는 EDI를 통하여 교환된 데이터를 업무 프로세스에 적용할 수 있는 개방적 구조를 가지기 때문에 업무 효율성 제고 등 실질적인 EDI 도입의 효과를 제공한다. 또한, 인터넷을 기반으로 하기 때문에 수많은 중소기업들이 구축 및 운영비용을 크게 걱정하지 않고 도입할 수 있다는 장점을 가진다[8].

XML/EDI의 구성요소는 다음의 <그림 2>와 같이 XML, EDI, 템플릿(template), 에이전트(agent), 저장소(repository)의 다섯 가지 기술 요소로 이루어져있으며, XML 기반의 메시지 교환시스템을 구성하는 기본적인 지침이 된다.



<그림 2> XML/EDI 구성요소

2.3 컴포넌트 개발방법

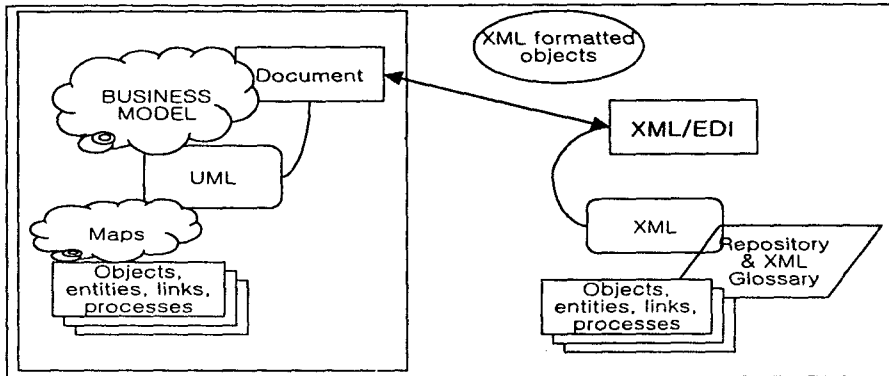
기업의 규모와 형태가 다변화되고, 업무 프로세스가 급변하고 있는 상황에서, 새로운 정보 기술과 더불어 기업 구조개선, 업무의 혁신 등을 주도할 체계적인 소프트웨어 개발 방법이 필요하게 되었는데, 이러한 요구를 충족시켜주는 객체지향 방법은 소프트웨어의 생산성과 품질을 획기적으로 제고하고 재사용성, 이식성, 확장성, 정보은닉, 호환성 등의 장점을 실현화 시켜주고 있다.

UML(Unified Modeling Language)은 Rumbaugh의 OMT(Object Modeling Technique) 방법론, Booch의 Booch 방법론, Jacobson의 OOSE(Object Oriented Software Engineering) 방법론과 기타 다른 전문가들의 주장을 통합하여 만든 모델링 개념의 공통집합으로 객체지향 분석과 설계영역에서 사용되는 표준화된 모델링 언어이다[7,9]. UML은 배우기 쉽고 확장하기도 쉬우며 다양한 표기법을 지니고 있어서 여러 방법론의 특징을 잘 표현할 수 있다. 분석단계에 만들어진 모형은 구현까지 연속성을 지니기 때문에 구현이나 설계에서의 변경은 곧 분석의 변경사항이 되어 반복적인 방법을 통해 시스템을 개발할 수 있다. 그리고 모델링에 대한 표현력이 강하고 비교적 모순이 적은 논리적인 표기법을 가져 시스템 규모에 상관없이 모두 적용이 가능하다.

UML 표준 언어를 이용한 컴포넌트는 표준화된 인터페이스를 바탕으로 독립적 재사용이 가능한 단위를 말한다. 컴포넌트를 보다 효율적으로 시험하기 위한 연구와 노력이 많이 진행되고 있다. Harrold(1999)는 컴포넌트 제공자와 컴포넌트 사용자의 관점에서 컴포넌트 기반 소프트웨어를 분석하여 시험하는 방법을 제시하였다[10]. Ghosh(1999)는 컴포넌트의 인

터페이스와 예외기준에 기반한 시험 방법을 제시하였다[11]. 국내에서도 단시간에 많은 컴포넌트를 시험하기 위한 시험 환경을 제안하였다[12].

이와 같이 대내외적인 연구에 따라 소프트웨어 산업계에서 컴포넌트 기반 프로그래밍에 대한 관심이 높아지고 보다 유연하고 확장성 높은 컴포넌트 기반의 아키텍처가 중요한 이슈로 등장하고 있다[1,6]. <그림 3>은 기업간의 상거래를 위한 문서를 처리할 경우, 객체지향의 UML 분석 및 설계를 통하여 XML/EDI 시스템과 상호 연계됨으로써 XML 문서를 전자적으로 처리하는 새로운 메시지교환시스템의 대안으로 제시되고 있다.

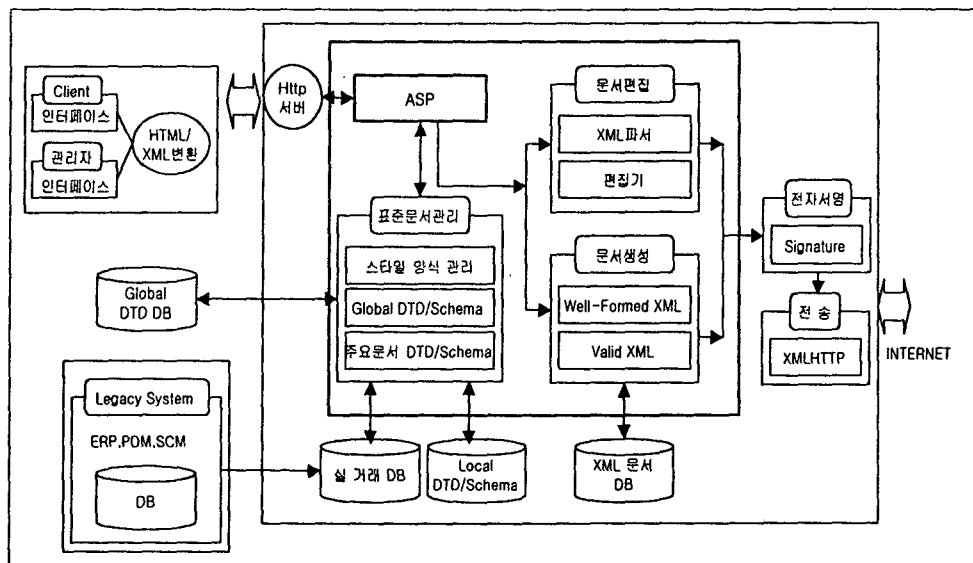


<그림 3> 객체지향 모델과 XML/EDI 연계

III. XML 메시지 교환 시스템의 구조

3.1 시스템 구조

본 논문에서는 XML 메시지 교환 시스템 구조를 <그림 4>와 같이 구성해보았다. 기존의 Legacy System과 연동해서 데이터를 가지고 표준문서인 Schema, DTD에 의해 표준 XML 문서를 생성하고 XSL을 이용하여 문서를 표현하고 전자서명을 거친 후 효율적인 전송을 위해 XMLHTTP를 이용해서 문서가 전송되어진다. 보다 정형화되고 표준화된 XML문서를 생성하기 위해서는 Schema나 DTD 설계에 보다 신중을 기해야 한다.



<그림 4> XML/EDI 시스템의 전체 구조

3.2 표준문서관리

기업간 상거래에 필요한 발주서, 납품서, 반품서, 검수서 등 실제 사용되는 주요 문서들에 대하여 문서 표준을 정의하고, 이를 XML DTD나 확장형 XML Schema로 정의한다. 또한 추가적으로 문서구조가 필요한 경우에는 확장이 가능하며, 국가 표준 DTD/Schema로 활용이 가능해진다. 표준문서관리 모듈에서 만들어진 DTD/Schema를 여러 사용자 사이에서 공유할 수 있도록 하고, 또한 데이터 처리와 정보교환에 대한 고려 및 발주, 납품과 관련된 중요한 정보의 사용자 간 이해를 위하여 양측에서 사용하는 EDI 태그의 의미를 일치시키기 위한 기능을 제공한다. 이때, 광역 저장소 내의 DTD/Schema는 거래상대방이 공유 가능하도록 한다.

3.3 문서편집

3.3.1 XML 파서(Parser)

XML 파서는 모든 XML 문서처리의 기반이 되는 핵심 기술이며, XML 문서를 해석하고 필요한 정보를 추출하며 오류를 점검하는 기능을 제공한다. 본 시스템에서는 XML 문서를 XML DOM(Document Object Model) 표준 인터페이스를 이용해서 구조와 내용을 어플리케이션에서 사용 가능하도록 만든다. 그리고 MS XML Parser와 W3C에서 제시하고 있는 표준인 DOM을 이용하여 XML Schema와 DI(Document Instance) 등 구성상의 오류를 점검하여 문서의 종류에 따른 계층모델과 인덱스 생성을 가능하게 한다.

3.3.2 XML 편집기

XML/EDI 문서 편집기는 XML 기반의 EDI 문서를 Visual하게 편집할 수 있는 응용 프로그램이다. XML Spy 편집기를 활용해서 Visual하게 편집을 한다. XML 문서의 편집은 크게 두 가지 형태로 구분해 볼 수 있다. DTD/Schema로 새로운 DI를 만드는 것과 DOM 트리를 이용해서 기존의 생성 문서를 편집하는 것이다.

3.4 문서생성

XML 문서 생성은 관계형 데이터베이스에서 DOM 트리를 이용하여 사전에 표준으로 정의된 DTD/Schema의 엘리먼트에 값을 전달함으로써 XML 파일이 생성된다. 문서생성모듈에서는 사용자가 데이터를 입력할 때 사용자 입력폼에 해당하는 문서의 Schema 정보를 가지고 있으며, DOM을 이용하여 메모리로 적재한 후 사용자 입력 폼에 정보를 제공한다.

3.5 XML/EDI 문서전송

유효한 XML 문서 생성후 사용자가 XML문서를 전송하고자 전송기능을 선택할 경우 전송될 대상기업에 대한 IP Address를 기존에 저장된 데이터베이스에서 가져와 해당 IP Address로 파일이 전송된다. 전송컴포넌트는 XML문서 전송, 수신 두 부분으로 나누어져 있으며, 메시지교환시스템의 메소드에 의해 XML 문서가 송수신된다.

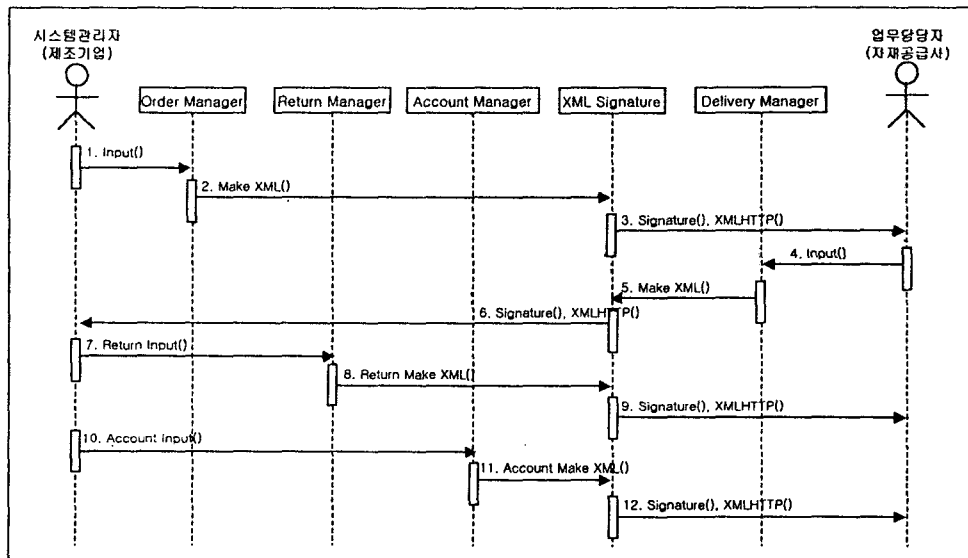
3.6 전자서명

공개키 암호방식에 기반을 둔 전자서명은 비밀키로 문서에 전자서명을 하고 공개키를 이용하여 이를 검증한다. 이러한 개인키 및 공개키의 성질을 이용하여, 문서를 수신한 상대방이 송신자의 신원을 확인하고, 문서의 위조, 변조를 방지하도록 한다. XML 전자서명 기능을 이용함으로써 메시지에 대한 인증(Authentication), 무결성(Integrity), 부인방지(Non-repudiation) 기능이 강화되어 안전한 메시지 교환이 가능해진다.

IV. 시스템의 설계 및 구현

4.1 객체지향 설계 모델

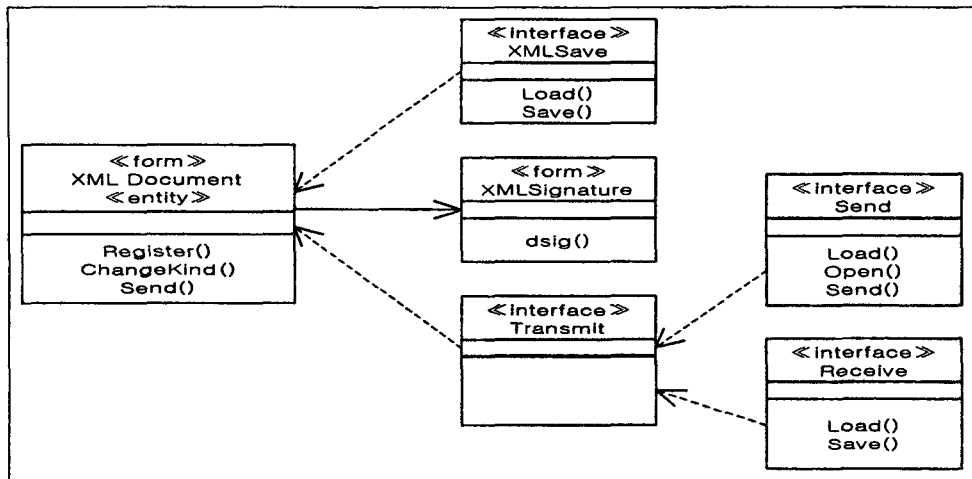
객체 설계단계에서는 분석단계에서 수행한 Use Case의 실제화를 검증하면서 개발 환경의 제약사항을 충분히 고려하여 메소드를 이용한 객체간의 정보전달을 구체화하게 된다. <그림 5>는 메시지 교환시스템을 구성하는 하위기능들에 대해 Sequence 다이어그램으로 표현한 것이다.



<그림 5> XML/EDI Sequence Diagram

4.2 Class Diagram

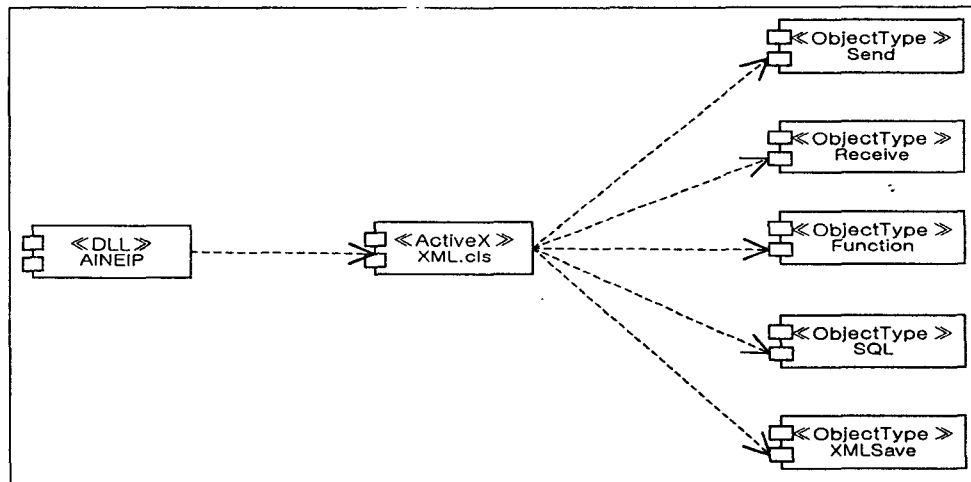
<그림 6>은 XML 메시지교환시스템의 업무객체를 Class 다이어그램으로 표현한 것이다. XML문서관리 <form>은 XMLSave, Transmit등의 <interface>에 의해서 XML 문서를 생성하고, 전송하게 된다. <form> 스테레오타입의 XMLSignature로 전자서명을 하고, Transmit <interface>에서는 전송(send), 수신(receive) 등의 <interface>가 있어서 XML 문서를 Load, Open, Send하는 하위 인터페이스를 가지고 있다.



<그림 6> XML/EDI Class Diagram

4.3 Component Diagram

Component 다이어그램은 전체적으로 개발되어야 할 컴포넌트의 수와 규모를 도식화한다. <그림 7>의 Component 다이어그램은 <DLL> 스테레오 타입을 가진 AINEIP 컴포넌트에서 XML 클래스의 전송(Send.bas), 수신(Receive.bas), 공용기능(Function.bas), 쿼리(SQL), XML문서저장(XMLSave.bas)등의 모듈을 도출하여 각 모듈간의 관계를 나타낸 것이다.



<그림 7> XML/EDI Component Diagram

4.4 XML 메시지 Schema

XML에 있어서 XML Schema는 XML의 구조를 논리적으로 설계하는 것이라고 할 수 있으며, 이 XML Schema를 이용하여 실제 내용이 들어가 있는 것을 XML 인스턴스 문서라고 한다[5]. <그림 8>은 발주서 Schema의 일부를 발췌한 내용으로 발주서의 구조를 정의하고 있다. 정의하는 방식으로 이 문서가 Schema 문서라는 것을 명시하고, 각각의 부모 Element와 자식 Element를 정의한 후 각각의 Type을 기술하고 있다. 이러한 Schema는 기업과 기업 간에 XML문서를 주고받을 경우 특정 Element가 어떤 형태의 데이터를 가지고 있는지, 해당 Element가 의미하는 것이 무엇인지 정확하게 전달할 수 있다. PurchaseOrder는 root 엘

리먼트로 시작해서 여러 개의 하위 엘리먼트를 구성되어 있다.

```

?xml version="1.0" encoding="UTF-8"?
<Schema xmlns="urn:schemas-microsoft-com:xml-data" xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="PurchaseOrder" content="eltOnly" order="seq">
    <element type="Head"/>
    <element type="Body" minOccurs="1" maxOccurs="*" />
  </ElementType>
  <ElementType name="Head" content="eltOnly" order="seq">
    <element type="OrderNumber"/>
    <element type="DeliveryDate"/>
    <element type="OrderDate"/>
    <element type="Sender"/>
    <element type="Customer"/>
  </ElementType>
  <ElementType name="OrderNumber" content="textOnly"/>
  <ElementType name="DeliveryDate" content="textOnly"/>
  <ElementType name="OrderDate" content="textOnly"/>
  <ElementType name="Sender" content="eltOnly" order="seq">
    <element type="SenderNumber"/>
    <element type="SenderName"/>
    <element type="SenderOwner"/>
    <element type="SenderAddress"/>
    <element type="SenderType"/>
    <element type="SenderKind"/>
    <element type="SenderTelephone"/>
    <element type="SenderFax"/>
  </ElementType>
</Schema>

```

<그림 8> XML 메시지 Schema

XML은 전자문서에 대한 스타일시트(XSL)와 문서구조 분석을 통한 문서형정의(DTD, Schema)를 개발함으로써 이를 기반으로 XML문서를 만들 수 있다. 단, 특정한 구조를 갖지 않는 문서일 경우에는 DTD나 Schema를 개발하지 않아도 된다.

일반적으로 Well-formed 문서라 함은 DTD나 Schema 없이 XML의 문법에 맞게 구성된 문서를 Well-formed 문서라고 한다. <그림 9>는 발주서 Well-formed XML문서의 일부분을 발췌한 것이다. XML문서에서 Sender 및 Customer의 정보, 문서의 본문에 해당하는 Body의 정보들은 DTD나 Schema 없이 작성된 문서이다. 또한 이들 데이터들은 XML 문법 상 아무런 이상이 없기 때문에 XML문서를 전송하고, Web Browser에 보여줄 때도 DTD나 Schema를 적용한 문서처럼 사용 가능하다.

```

<?xml version="1.0" encoding="euc-kr"?>
<PurchaseOrder xmlns="x-schema:Delivery.xdr">
  <Head>
    <OrderNumber>20020516-A56D52</OrderNumber>
    <DeliveryDate>20020522</DeliveryDate>
    <OrderDate>2002년 5월 16일</OrderDate>
    <Sender>
      <SenderNumber>505-81-05168</SenderNumber>
      <SenderName>(주)광진상공</SenderName>
      <SenderKind>자동차부품업</SenderKind>
      <SenderTelephone>(02 )679 -5264</SenderTelephone>
    </Sender>
    <Customer>
      <CustomerNumber>504-81-37629</CustomerNumber>
      <CustomerName>동국정밀(주)</CustomerName>
      <CustomerKind>자동차부품</CustomerKind>
      <CustomerTelephone>053-353-3413</CustomerTelephone>
    </Customer>
  </Head>
  <Body>
    <PartNumber>82436-38001MO</PartNumber>
    <MatrialName>HOLDER-UPR TUBE, LH, 사출</MatrialName>
    <OrderQuantity>4,430</OrderQuantity>
    <UnitCost>47</UnitCost>
    <OrderAmount>208,210</OrderAmount>
    <OrderSum>208,210</OrderSum>
  </Body>
</PurchaseOrder>

```

<그림 9> Well-formed XML 문서

4.5 XML 문서생성 컴포넌트

<그림 10>은 Component Diagram에 의해서 도출된 Component 중 XML 문서생성을 위한 Web form 입력 메소드를 코드화한 것이다. 아래의 소스코드는 웹 폼에서 입력되는 데이터값을 KindSQL()이라는 공통모듈로 전달하는 역할을 한다. OrderXML이라는 메소드는 모든 발주 Component에서 공통으로 사용할 수 있게 하여 코드의 재사용성을 높였다.

```
Public Sub OrderXML(SearchKEY, UID, Kind, PNO_Condition As Variant)
    Set vanconn = New ADODB.Connection
    Set Rs = New ADODB.Recordset
    vanconn.CursorLocation = adUseClient
    Dim SQL, recordcount As String
    vanconn.Open strconn
    SQL = KindSQL(SearchKEY, UID, Kind, PNO_Condition)
    Rs.Open SQL, vanconn
End Sub
```

<그림 10> XML 문서생성을 위한 Parameter전달 소스

4.6 XML 메시지 전송 컴포넌트

4.6.1 XML 메시지 전송

전송Component는 XML문서 전송, 수신 두 부분으로 나누어져 있다. 각 Component는 DOMDocument 4.0과 ServerXMLHTTP 4.0을 사용하여 XML 파일을 송수신한다. [표 1]에서는 생성된 XML문서를 전송하기 위해 XML파일을 load()메소드를 통해서 읽어들이고, 읽어들이는 XML데이터들은 전송하기 전 ServerXMLHTTP 객체를 open() 메소드를 통해서 활성화시킨 후 send()메소드를 통해서 전송한다.

[표 1] 주요Class 및 메소드 설명

사 용 객 체	메소드명	설 명
DOMDocument 4.0	load()	XML문서를 전송하기 위해 XML파일을 읽어옴
ServerXMLHTTP 4.0	open()	문서를 전송하기 전 ServerXMLHTTP 객체를 활성화.
	send()	XML문서를 전송.

4.6.2 XML 메시지 수신

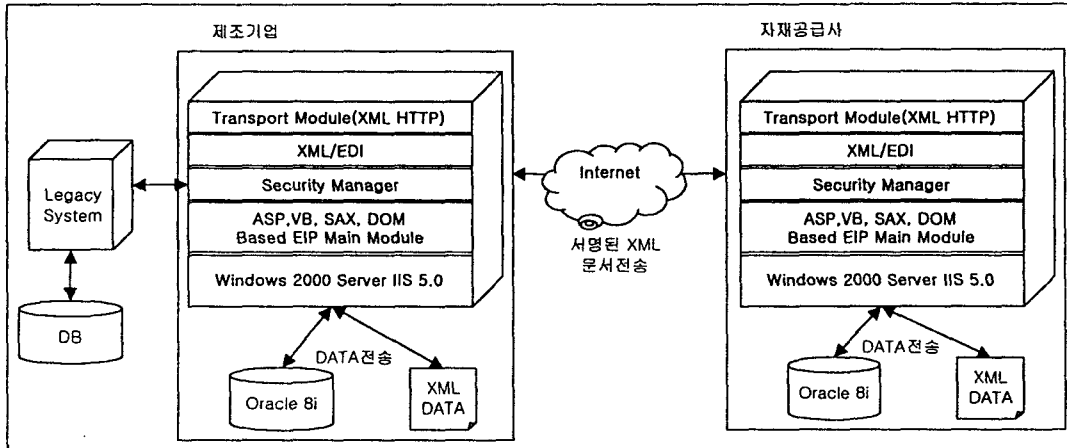
[표 2]에서는 전송되어진 XML데이터를 수신하는 모듈로써, load() 메소드와 save() 메소드가 사용된다. load() 메소드는 서버측에서 전송되어진 XML데이터를 받아오는 역할을 수행하며, save() 메소드는 받아온 XML데이터를 저장하는 역할을 수행한다.

[표 2] 주요Class 및 메소드 설명

사 용 객 체	메소드명	설 명
DOMDocument 4.0	load()	서버측에서 전송된 XML 스트림을 load()메소드를 통해 받음.
	save()	받아온 XML 스트림을 XML파일로 저장.

4.7 XML/EDI 구성도

본 시스템의 서버는 Windows 2000 Server 플랫폼을 기반으로 Oracle 8i DBMS를 채택하였으며, Web Browser는 Internet Explorer 6.0을 활용하여 Web 환경에서 전자문서를 교환하고 있다. 기업의 Legacy System에서 제공된 데이터를 PL/SQL 프로시저로 XML/EDI 서버의 DB에 저장하고, 저장된 데이터를 ASP3.0과 XML DOM을 이용해서 읽어들이어서 XML 문서를 생성하고 생성된 XML문서는 XSL, XSLT를 이용해서 출력하거나 전송할 수 있는 시스템으로 구성된다.



<그림 11> XML/EDI 구성도

4.8 구현 화면

4.8.1 발주관리 화면 구성도

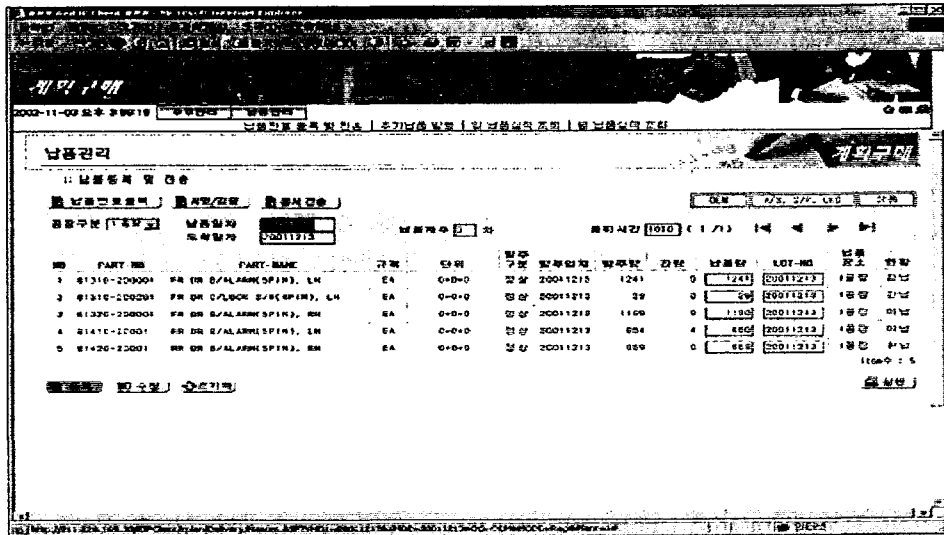
<그림 12>는 제조기업 업무담당자가 인증을 거친 후 발행년월일과 업체코드를 넣고 XML생성 버튼을 클릭하면 XML문서가 특정 폴더에 저장된다. 파일이름은 발주구분, 연월일, 시간, 업체코드로 (TORDER_20020917_15_25_16_100070.xml)구성이 된다. 이렇게 생성된 파일을 전자서명을 이용해서 서명을 하면 기존의 파일 이름에 .xmlSIGN이 붙고 확장자 .XML이 붙는다(TORDER_20020917_15_25_16_100070.xmlSIGN.XML). 이렇게 전자서명된 파일을 XML 전송 버튼을 눌러 자재공급사에 전송된다.

No	발주구분	PART-NO	PART-NAME	단위	금액	12.15월 06:00 현재	01월	02월	03월	04월	05월	06월	07월	08월	09월	10월	11월	12월		
1	1종장	81310-20001	FR DR B/ALARMSPRL LH	EA	0+0	1214	3238	-24	1287	1217	768	471	878	381	270	0	0	793	1076	
2	1종장	81310-20001	FR DR CLOCK ACTIPRL LH RND	EA	0+0	159	1	150	0	150	20	150	0	0	0	0	0	0	0	
3	1종장	81310-20001	FR DR CA/OC SWMSPRL LH	EA	0+0	731	302	409	0	436	208	282	121	114	0	0	0	0	189	227
4	1종장	81310-20001	FR DR B/LATCHLATCH ONLV3 LH	EA	0+0	171	29	149	0	149	26	119	0	0	0	0	0	0	0	
5	1종장	81320-20001	FR DR B/ALARMSPRL LH	EA	0+0	1274	3368	0	1180	1217	768	471	878	381	270	0	0	793	1076	
6	1종장	81320-20001	FR DR CLOCK ACTIPRL LH	EA	0+0	712	130	582	0	392	221	261	207	121	80	0	0	0	200	679
7	1종장	81320-20001	FR DR CA/OC SWMSPRL LH RND	EA	0+0	159	25	183	0	159	0	159	0	0	0	0	0	0	26	9
8	1종장	81320-20001	FR DR B/LATCHLATCH ONLV3 LH	EA	0+0	165	20	158	0	150	20	135	0	0	0	0	0	0	0	
9	1종장	81310-20001	FR DR B/ALARMSPRL LH	EA	0+0	1036	3182	353	886	1217	771	448	800	381	248	0	0	799	1126	
10	1종장	81310-20001	FR DR CA/OC ACTIPRL LH	EA	0+0	988	159	808	0	888	238	650	158	121	80	0	0	0	276	311
11	1종장	81410-20001	FR DR B/LATCHLATCH ONLV3 LH	EA	0+0	207	26	181	0	181	0	181	0	20	0	0	0	0	0	

<그림 12> 발주서 작성 및 XML 파일 생성

4.8.2 납품관리 구현 화면

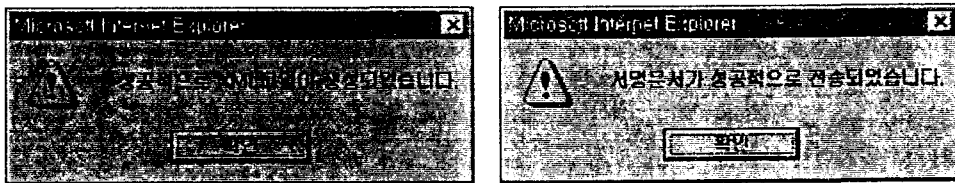
<그림 13>은 자재공급사 메인 화면으로 발주 조회하는 수주관리와 발주에 따른 납품서 등록을 하는 납품관리 메뉴로 구성된다. 수주관리에는 일일발주서 조회, 주간 소요계획 조회, 월간소요계획 조회, 추가발주서 조회로 구성되며 납품관리에는 납품전표등록 및 전송, 추가납품 발행, 일납품 내역, 월납품실적 조회로 구성되어 있다.



<그림 13> 납품서 작성 및 XML 파일 생성

<그림 13>은 자재 공급사의 업무 담당자가 발주조회에 따른 납품서 등록을 하는데 납품 일자(발주 일자)는 자재공급사에서 발주한 날짜이고, 도착일자는 납품이 도착하는 예정일이다. 납품 차수는 같은 날 납품을 하는 횟수이며 출하시간으로 구성되어져서 등록을 하면 XML 파일이 생성된다. 자재공급사 업무담당자는 생성된 XML 파일을 전자서명 거친 후 제조기업으로 문서를 전송한다.

<그림 14>는 XML 파일이 성공적으로 생성되고, 서명된 XML문서가 성공적으로 전송되었다는 것을 확인하는 팝업 창이다. 이 기능은 XML문서를 XML 전자서명 기능에 의해 암호화함으로써 메시지의 인증, 무결성, 부인방지 기능을 강화하여 안전한 메시지 교환이 가능하도록 한다.



<그림 14> XML 생성 및 전송 확인 팝업창

4.9 분석 및 평가

본 시스템의 특징은 XML 메시지교환시스템의 구현에 있어 UML에 의한 분석 및 설계 그리고 컴포넌트기반의 개발로 연계되어 구축됨으로써 소프트웨어의 재사용성과 정보은닉, 그리고 상호운영성을 높였다는 것이다. 즉, 컴포넌트기반으로 개발함으로써 반복되는 프로그램의 재사용이 가능하고, Component만 호출하여도 그 기능을 수행할 수 있도록 하였다.

또한 핵심 모듈 또는 중요 소스코드에 대해서도 Component로 작성하여 컴파일 과정을 거침으로써 소스코드에 대한 정보은닉의 효과가 있으며, 일단 Component로 만들어진 모듈은 이기종의 플랫폼에 상관없이 사용가능하기 때문에 높은 이식성을 가지고 있다.

이외에도 데이터와 표현양식이 분리되어 있는 XML을 사용함으로써 사용자 특성에 맞는 다양한 표현양식을 사용할 수 있도록 하였다. 과거에는 데이터와 표현양식이 매우 밀접하게 결합되어 동일한 데이터를 다른 형태로 변환하려면 새로운 표현양식을 만들어야 했다. 하지만 XML을 사용하므로 이러한 부담을 많이 간소화시킬 수 있었다.

그리고 기존의 EDI에서는 전송되는 메시지 중에서 데이터 항목을 분리하고, 식별하기 위해서 독특한 세그먼트 식별자를 사용함으로써 사용과정에서 수정이 힘들었는데 반해, XML 메시징 시스템에서는 이러한 세그먼트 식별자를 XML Schema로 정의하기 때문에 수정이 용이하다.

또한 기존의 EDI문서 전송에서는 VAN을 이용한 전송방법을 수행하였지만, 본 시스템의 XML 메시징시스템에서는 XMLHTTP라는 객체를 이용함으로써 인터넷상에서 쉽게 XML 문서를 전송할 수 있도록 하였다.

본 논문에서 개발한 컴포넌트기반의 XML 메시지교환시스템을 국내에서 개발한 타 시스템과 비교할 경우, [표 3]과 같이 비교할 수 있다[3, 5]. [표 3]에서 보듯이 본 시스템은 객체 지향분석 및 설계기법으로 UML을 활용하여 컴포넌트 기반으로 개발하였으며, 전송방법에서도 타 시스템은 SMTP, POP3를 사용했지만 본 시스템은 XMLHTTP 객체를 사용해서 보다 간결하고 쉽게 문서를 전송 할 수 있다. 그리고 DOM 객체를 활용하여 DB와의 연동을 용이하게 함으로써 기업에서 활용하는 ERP, SCM, CRM과 같은 Legacy System과의 호환성이 높고, 상호 연동이 가능하다.

[표 3] 타 시스템과의 비교

구 분	본 시스템	타시스템[3]	타 시스템[5]
분석방법	UML	미적용	미적용
CBD 적용여부	적용	미적용	미적용
표준문서	Schema, DTD	DTD	DTD
전송방법	HTTP(XMLHTTP)	SMTP, POP3	SMTP, POP3
Legacy System 호환	상호 연동(ERP)	독립적 운영	독립적 운영.
웹기반의 사용자, 관리자 모듈	웹기반	웹기반	독립 응용프로그램
XML문서생성	HTML 입력폼	HTML 입력폼	사용자의 엘리먼트 선택에 따르는 문서생성
개발운영체제	Windows 2000 Server	리눅스 6.2	Windows NT 4.0
표준 API	MSDOM	JDOM	MSDOM
문서표현	XSL, CSS, HTML	XSL	XSL

V. 결론

XML/EDI 기술은 차세대 EDI의 구현 기술로 자리를 잡아가고 있다. 이 기술은 차세대 전자문서의 표준으로 받아들여지는 XML기술을 EDI 메시지 교환시스템에 적용함으로써 전통적인 EDI에서 처리할 수 있는 업무의 한계를 벗어나 전자상거래 전반에서 필요한 프레임워크를 제공하는 것이라 할 수 있다. 즉 XML을 이용하여 타 업무 프로세스와 연계될 수 있으며, 인터넷을 기반으로 한다는 점에서 VAN EDI에 비해 상대적으로 훨씬 저렴한 비용으로 구축, 유지될 수 있다[4].

본 논문에서 구현한 XML 메시징 시스템은 객체지향 방법론인 UML을 이용하여 제조기업과 자재공급사간의 발주, 납품 처리 등의 업무에 적용해 보았다. 또한 객체지향 방법인 UML로 기능을 분석하였고, 확장 가능한 표준문서를 Schema를 이용하여 작성하였다. 전송 방법으로 기존의 XML/EDI 시스템은 SMTP, POP3를 이용했지만 본 시스템은 XMLHTTP를 사용했고, Legacy System과의 호환성, 웹기반의 모듈 보안성에 대하여 구현하였다.

UML을 이용한 XML 메시징 시스템의 분석 및 설계를 통하여 전체적인 Use Case 다이어그램을 기술하고, 그에 맞는 Sequence 다이어그램을 표현하였다. 그리고 분석된 요구사항 및 모델을 기반으로 하여 Class 다이어그램과 실제 구현 환경에 적용하기 위한 객체구현모델을 Component 다이어그램으로 도출하여 재사용성을 높였다.

본 시스템을 업무에 적용함으로써 기업간(B2B) 전자문서 거래 처리시간 단축, 업무 부대비용 감소, 업무오류 방지, 물류비 절감, 인력절감 등의 효율성 증대를 통하여 조직의 경쟁력을 향상시키고 전자상거래 활성화에 크게 기여 할 것이라 사료된다.

본 시스템의 개발과정에서 나타난 한계점으로는 첫째, 대상업무의 업무표준이 아직 정립되지 않아서 사실 표준을 사용하였으며, 산업적 적용을 위해서는 문서의 표준화가 필요하다는 점이다. 둘째, XML 메시징 시스템의 구현에서 VAN/EDI시스템을 수용하는 메커니즘이 구현되지 않아 상호 연동이 안된다는 점이다. 그리하여 향후 연구방향으로는 본 시스템의 적용을 위해 전자문서 표준화에 대한 작업이 필요하며, ebXML과 같은 표준안을 활용하여 구체화하는 연구가 진행될 수 있다. 또한 SOAP를 통해 RPC(Remote Procedure Call)를 호출하여 이기종간의 메시지를 전송하기 위한 연구가 필요하다. 그리고 전통적인 EDI를 수용할 수 있는 변환모듈에 대한 연구를 통하여 이를 확대 적용하는 연구가 수행될 수 있다.

참고문헌

- [1] 김준산, 강제성, 마유승, 권용래, XMI로 표현된 UML 모델로부터 생성된 시험 데이터를 이용한 컴포넌트 기반 소프트웨어 시험 도구, 정보과학회지, 2002
- [2] 류형규, 이순천, 류시원, 신성호, UML 기반 객체지향 클라이언트/서버 구축, 홍릉과학출판사, 2000
- [3] 신동규, 신동일, XML/EDI 시스템의 설계 및 구현, 2001
- [4] 심중석, 정경진, 전자상거래와 e-비즈니스, 청림출판, 2000
- [5] 임영태, 한우용, 정희경, XML에 기반한 EDI 문서교환 시스템 설계 및 구현, 2000
- [6] 윤용진, 김병기, 배종성, UML 기반의 웹 애플리케이션 향해 개념모델 연구, 한국정보처리학회지, 제9권, 제1호, 2002
- [7] 윤청, 소프트웨어 공학, 생능출판사, 2000
- [8] 한국전산원, SGML, XML, EDI 통합 및 연계방안, 1999
- [9] G.Booch, I.Jacobson, and J. Rumbaugh, The Unified Modeling Language User

Guide, The Addison-Wesley

[10] M.J.Harrold, D. Liang, and S. Sinha "An approach to analyzing and testing component-based systems" In Proceedings of the 1st International ICSE Workshop on Testing Distributed Component-Based Systems, 1999.

[11] S. Ghosh and A.P. Mathur "Issues in testing distributed component-based systems" In Proceedings of the 1st International ICSE Workshop on Testing Distributed Component-Based Systems, 1999.

[12] Y.S.Ma, S.U.Oh, D.H.Bae and Y.R.Kwon "Framework for Third Party Testing of Component Software" In the Proceedings of 8th Asian-Pacific Software Engineering Conference, 2001.